

Implementing ANZSRC (2008) in Eprints at the University of Tasmania

June/July 2008

Mr. Christian McGee
C.McGee@utas.edu.au
School of Computing and Information Systems
University of Tasmania

1. Background

On the 31st of March 2008, the Minister for Innovation, Industry, Science and Research, Senator Kim Carr announced a new classification scheme to replace the 1998 ASRC: Australian and New Zealand Standard Research Classification (ANZSRC). The Minister has said it “will be a primary tool for defining disciplines and the make-up of discipline clusters in the Government’s Excellence in Research for Australia (ERA) initiative”. For further information see Senator Carr’s media announcement:

<http://www.minister.industry.gov.au/SenatortheHonKimCarr/Pages/NEWCLASSIFICATIONTOASSISTERADEVELOPMENT.aspx>

The new classification replaces the three old tables RFCD, TOR and SEO with three updated tables. The TOR is basically exactly the same but has become “Type of Activity” or TOA. The RFCD has been renamed to “Fields of Research” or FOR and has been revised to accommodate new fields and New Zealand. Although the 2008 version of the “Socio-Economic Objective” or SEO has not been renamed it has also been revised to accommodate new objectives and New Zealand. The full ANZSRC classification is available from the Australia Bureau of Statistics:

<http://www.abs.gov.au/AUSSTATS/abs@.nsf/Latestproducts/1297.0Contents12008?opendocument&tabna>

In June/July of 2008 the University of Tasmania implemented the ANZSRC into its Eprints digital repository software. This document describes how this implementation was achieved. Source code and related files are available for download from the following URL: <http://eprints.utas.edu.au/7002/>

This implementation was successfully completed on both Eprints version 3.03rc1 running under Solaris 10 and Eprints 3.05 running under Red Hat Linux Enterprise Server 5.

2. Eprints Database Changes

Two new Eprints MySQL tables are required to implement the new for08 and seo08 subjects. File listings (A1 and A2) of the text files used to create these MySQL tables are included in Appendix A.

To execute the statements contained in these files, execute the following commands on your Eprints database host (in the case of UTas the Eprints application and database are installed on the same host):

```
mysql -u <your_eprints_dbuser> -p <your_eprints_dbname> < for08.sql
mysql -u <your_eprints_dbuser> -p <your_eprints_dbname> < seo08.sql
eg: mysql -u ep3dbuser -p ep3db < for08.sql
mysql -u ep3dbuser -p ep3db < seo08.sql
```

3. Producing and Importing Eprints XML Subject Files

An excel spreadsheet was downloaded from RadWiki:

<http://www.radwiki.edu.au/images/0/0e/JCU-FORandSEO2008mapping.xls>

This was a good starting point but many errors and omissions were noticed. As a result this spreadsheet was converted into two text files (FOR2008.txt and SEO2008.txt) that are hopefully a more accurate representation of the ANZSRC 2008. To eliminate errors, a time consuming process of comparing printouts of the official pdf version of the ANZSRC 2008 and these two text files was undertaken before the files were used.

Two basic perl scripts were written to convert these text files to Eprints XML suitable for importing into Eprints. The program listings (B1 and B2) are available in Appendix B. The XML files produced by these scripts were imported into Eprints using the following commands, executed as the Eprints unix user:

```
~/bin/import_subjects epprod --xml --nopurge --force FOR2008.xml
~/bin/import_subjects epprod --xml --nopurge --force SEO2008.xml
```

Prior to executing these commands it is good practice to make a backup of the Eprints subjects:

```
~/bin/export epprod subject XML > subjects_bup.xml
```

If there are problems it will be possible to restore your original subjects by executing the following command:

```
~/bin/import_subjects epprod --xml --force subjects_bup.xml
```

4. Eprints Configuration Changes

The following instructions are specific to the UTas Eprints server. However, the steps below should be fairly similar on other Eprints 3.0.x implementations.

4.1 Workflow

Edit Eprints workflow so that ONLY the new FOR and SEO are available for deposit by normal users. Admins and editors should also be able to edit the old subjects and seo codes as well as collections.

```
vi <path_to_eprints>/archives/<your_archive_name>/cfg/workflows/eprint/default.xml

<?xml version="1.0" encoding="utf-8"?>

<workflow xmlns="http://eprints.org/ep3/workflow"
xmlns:epc="http://eprints.org/ep3/control">

  <flow>
    <stage ref="type"/>
    <stage ref="files"/>
    <stage ref="pubinfo" />
    <stage ref="for08" />
    <stage ref="seo08" />
    <epc:if test="$STAFF_ONLY = 'TRUE'">
      <stage ref="subjects" />
      <stage ref="seos" />
      <stage ref="collections" />
    </epc:if>
  </flow>

  ... stuff deleted ...

  <stage name="for08">
    <component type='Field::Subject'><field ref="for08" /></component>
  </stage>

  <stage name="seo08">
    <component type='Field::Subject'><field ref="seo08" /></component>
  </stage>

  <stage name="subjects">
    <component type='Field::Subject'><field ref="subjects" required="yes" /></component>
  </stage>
```

```

<stage name="seos">
  <component type='Field::Subject'><field ref="seos" /></component>
</stage>

<stage name="collections">
  <component type='Field::Subject'><field ref="collections" /></component>
</stage>
</workflow>

```

4.2 Phrases

```

cd <path_to_eprints>/archives/<your_archive_name>/cfg/lang/en/phrases
vi eprints_fields.xml

```

```

<epp:phrase id="eprint_fieldname_seo08">SEO 2008</epp:phrase>
<epp:phrase id="eprint_fieldhelp_seo08">Please select one Socio-Economic Objective
(SEO2008) classification from the list below. Then click 'Add'.</epp:phrase>
<epp:phrase id="eprint_fieldname_for08">Fields of Research 2008</epp:phrase>
<epp:phrase id="eprint_fieldhelp_for08">Please select at least one main Fields of
Research (FOR2008) category, and optionally up to two other FOR2008 categories you think
are appropriate for your submission, from the list below. Then click 'Add'.</epp:phrase>

```

```
vi workflow.xml
```

```

<epp:phrase id="metapage_title_for08">FOR2008</epp:phrase>
<epp:phrase id="metapage_title_seo08">SEO2008</epp:phrase>

```

```
vi views.xml
```

```

<epp:phrase id="viewname_eprint_subjects">Subject</epp:phrase>
<epp:phrase id="viewname_eprint_for08">Fields of Research (ANZSRC 2008)</epp:phrase>
<epp:phrase id="viewname_eprint_types">Type</epp:phrase>
<epp:phrase id="viewname_eprint_seos">Socio-Economic Objective</epp:phrase>
<epp:phrase id="viewname_eprint_seo08">Socio-Economic Objective (ANZSRC 2008)</epp:phrase>
<epp:phrase id="viewname_eprint_collections">Collections</epp:phrase>

```

4.3 Fields, Views, Abstracts and Search configs

```

cd <path_to_eprints>/archives/<your_archive_name>/cfg/cfg.d
vi eprint_fields.pl

```

```

{
  'top' => 'for08',
  'render_input' => undef,
  'browse_link' => 'for08',
  'name' => 'for08',
  'multiple' => 1,
  'type' => 'subject'
},
{
  'top' => 'subjects',
  'render_input' => undef,
  'browse_link' => 'subjects',
  'name' => 'subjects',
  'multiple' => 1,
  'type' => 'subject'
},
{
  'top' => 'collections',
  'render_input' => undef,
  'browse_link' => 'collections',
  'name' => 'collections',
  'multiple' => 1,
  'type' => 'subject'
},
{
  'top' => 'seo08',
  'render_input' => undef,
  'browse_link' => 'seo08',
  'name' => 'seo08',
  'multiple' => 1,
  'type' => 'subject'
},

```

```

    {
        'top' => 'seos',
        'render_input' => undef,
        'browse_link' => 'seos',
        'name' => 'seos',
        'multiple' => 1,
        'type' => 'subject'
    },

vi eprint_render.pl

# FOR08s...
$stable->appendChild( $session->render_row(
    $session->html_phrase( "eprint_fieldname_for08" ),
    $eprint->render_value( "for08" ) ) );

# Subjects...
$stable->appendChild( $session->render_row(
    $session->html_phrase( "eprint_fieldname_subjects" ),
    $eprint->render_value( "subjects" ) ) );

# SEO08s...
$stable->appendChild( $session->render_row(
    $session->html_phrase( "eprint_fieldname_seo08" ),
    $eprint->render_value( "seo08" ) ) );

# SEOs...
$stable->appendChild( $session->render_row(
    $session->html_phrase( "eprint_fieldname_seos" ),
    $eprint->render_value( "seos" ) ) );

# Collections...
$stable->appendChild( $session->render_row(
    $session->html_phrase( "eprint_fieldname_collections" ),
    $eprint->render_value( "collections" ) ) );

vi search.pl

$c->{search}->{advanced} =
{
    search_fields => [
        { meta_fields => [ $EPrints::Utils::FULLTEXT ] },
        { meta_fields => [ "title" ] },
        { meta_fields => [ "creators_name" ] },
        { meta_fields => [ "abstract" ] },
        { meta_fields => [ "keywords" ] },
        { meta_fields => [ "for08" ] },
        { meta_fields => [ "subjects" ] },
        { meta_fields => [ "seo08" ] },
        { meta_fields => [ "seos" ] },
        { meta_fields => [ "collections" ] },
    ],

vi views.pl

{
    id => "seos",
    fields => "seos",
    order => "-date/title",
    hideempty => 1,
},
{
    id => "for08",
    fields => "for08",
    order => "-date/title",
    hideempty => 1,
},
{
    id => "seo08",
    fields => "seo08",
    order => "-date/title",
    hideempty => 1,
},

```

4.4 Eprints Home Page

```
cd <path_to_eprints>/archives/<your_archive_name>/cfg/lang/en/static
vi index.xpage

<td class="box_padding"></td>
<td valign="top" class="box_padding"><ul>
  <li><a href="/view/authors/">by author</a></li>
  <li><a href="/view/year/">by year </a></li>
  <li><a href="/view/for08/">by field of research (ANZSRC 2008)</a></li>
  <li><a href="/view/se08/">by socio-economic objective (ANZSRC 2008)</a></li>
  <li><a href="/view/subjects/">by subject (ASRC 1998)</a></li>
  <li><a href="/view/seos/">by socio-economic objective (ASRC 1998)</a></li>
  <li><a href="/view/collections/">by collection</a></li>
  <li><a href="/perl/latest">by latest additions</a> </li>
</ul></td>
<td class="box_padding">&nbsp;</td>
</tr>
```

4.5 Eprints Header

```
cd /d/app/eprints3/archives/epprod/cfg/static/javascript/auto
vi eprints.js

window.mm_menu_0821133021_1 = new Menu("root",220,18,"Verdana, Arial, Helvetica, sans-serif",12,"#ffffff", "#ffffff", "#999999", "#cc240c", "left", "middle", 3,0,1000,-5,7,true,true,true,0,true,true);
mm_menu_0821133021_1.addMenuItem("by&nbsp;author", "location='http://eprints.utas.edu.au/view/authors/'");
mm_menu_0821133021_1.addMenuItem("by&nbsp;year", "location='http://eprints.utas.edu.au/view/year/'");
mm_menu_0821133021_1.addMenuItem("by&nbsp;field of research (2008)", "location='http://eprints.utas.edu.au/view/for08/'");
mm_menu_0821133021_1.addMenuItem("by&nbsp;socio-economic objective (2008)", "location='http://eprints.utas.edu.au/view/se08/'");
mm_menu_0821133021_1.addMenuItem("by&nbsp;subject (1998)", "location='http://eprints.utas.edu.au/view/subjects/'");
mm_menu_0821133021_1.addMenuItem("by&nbsp;socio-economic objective (1998)", "location='http://eprints.utas.edu.au/view/seos/'");
mm_menu_0821133021_1.addMenuItem("by&nbsp;collection", "location='http://eprints.utas.edu.au/view/collections/'");
mm_menu_0821133021_1.addMenuItem("by&nbsp;latest&nbsp;additions", "location='http://eprints.utas.edu.au/perl/latest'");

~/bin/generate_static <your_archive_name>
cd <path to eprints>/archives/<your_archive_name>/html/en/
vi index.title
(remove the text "Home")
```

4.6 TOR renamed to TOA

The extra field required to implement the Type of Research (TOR) had already been implemented as part of the modifications required for the HERDC 2007 import implemented in March/April 2008. Details are available at:

<http://eprints.utas.edu.au/7003/>

```
cd <path_to_eprints>/archives/<your_archive_name>/cfg/lang/en/phrases
vi eprints_fields.xml
(change TOR to TOA and "Type of Research" to "Type of Activity")
```

4.7 Regenerating abstracts and views

To make sure all the changes are loaded into Eprints, restart Apache.
As the eprints unix user:

```
~/bin/generate_views <your_archive_name>
~/bin/generate_abstracts <your_archive_name>
```

5. Mapping old ASRC (1998) classifications to new ANZSRC (2008) classifications

Most (if not all) existing ePrints will have OLD ASRC classifications assigned. The old 1998 RFDC and SEO codes will need to be mapped to the new 2008 FOR and SEO codes. An Excel spreadsheet containing the mappings is available for download at:

<http://www.abs.gov.au/AUSSTATS/abs@.nsf/DetailsPage/1297.02008?OpenDocument>

The mappings are also described in the official ANZSRC 2008 specification document also available for download (in PDF) from the above URL. The Excel spreadsheet was used to create suitable mapping text files for later processing by a set of Perl scripts developed in-house.

5.1 Assumptions and Assertions

1. One to one mappings can be done automatically:
FOR1998 --> FOR2008
 - eg: 230117 --> 010206SEO1998 --> SEO2008
 - eg: 620101 --> 820507
2. One to many mappings will need to be done manually:
FOR1998 --> FOR2008-1, FOR2008-2, , FOR2008-n
 - eg: 240502 --> 010502, 010503, 020303SEO1998 --> SEO2008-1, SEO2008-2, , SEO2008-n
 - eg: 779905 --> 960501, 960506, 960599
3. Any required manual mappings should be identified during the automatic mapping process.
4. On July 3rd, the UTas Eprints working party agreed not to automatically map any “Other” SEO or FOR classifications. Any classification number ending in “99” is considered to be an “Other” classification. This includes mappings of the type:
FOR1998 (other) --> FOR2008 (other)
 - eg: 380299 --> 200499FOR1998 (other) --> FOR2008 (non-other)
 - eg: 370399 --> 160101FOR1998 (non-other) --> FOR2008 (other)
 - eg: 300509 --> 070799SEO1998 (other) --> SEO2008 (other)
 - eg: 769999 --> 969999SEO1998 (other) --> SEO2008 (non-other)
 - eg: 660399 --> 850605SEO1998 (non-other) --> SEO2008 (other)
 - eg: 749901 --> 939999
5. Initially and for some time after implementation it will be necessary to have both classifications (ANZSRC 2008 and ASRC 1998) installed. The old RFCD and SEO should not be able available for selection by normal Eprints users after installation of the new FOR and SEO codes. New Eprints deposited should use the new classification and new codes. The old codes should be available in the Eprints workflow for admin users ONLY. Some time in the future, once the manual mapping has been completed, the old codes could be removed from the system.

5.2 1998 to 2008 Mapping Implementation

The Excel spreadsheet “1297.0 correspondence tables.xls” was used to produce two text files (for2008.csv and seo2008.csv) containing just comma separated RFCD / FOR and SEO 98 / SEO2008 codes.

| Extract from for2008.csv RFCD,FOR08 | Extract from seo2008.csv SEO98,SEO08 |
|--|---|
| 230101,010104p | 610101,810103p |
| 230101,010107p | 610101,810105p |
| 230101,010506p | 610101,810106p |
| 230102,010101p | 610101,810108 |
| 230103,010101p | 610101,810109p |
| 230104,010103 | 610102,810102 |
| 230105,010105 | 610102,810103p |

A set of perl scripts were developed to read in these files, do the automatic “one to one” mapping, and report on the eprintids and codes for mappings that must be done manually. The scripts developed also produce a list of eprintids and RFCD codes that were incorrectly set as “depositable”. These “not-depositable” RFCD codes end in “00” and do not have FOR08 equivalents in the correspondence tables. See Appendix C for program listings. To use these scripts you will need to add your Eprints database details to the Settings.pm file. You can check what the scripts will do by just running them – by default they will not make any database changes. There are two scripts to run:

```
./for1998-2-2008.pl  
./seo1998-2-2008.pl
```

Once you have checked the output from these commands and are satisfied with the results, modify the Settings.pm file so that the \$UPDATEDDB variable is set to “1”.

Rerun the commands, redirecting the output to a file, so that the results can be viewed and acted upon at a later date:

```
./for1998-2-2008.pl > for.out  
./seo1998-2-2008.pl > soe.out
```

The for.out and soe.out files will contain details of the mappings that were made automatically and the mappings that need manual intervention.

The new FOR and SEO codes will not be immediately visible on the eprint abstract pages. The abstract pages are static and must be regenerated by executing the following command, as the Eprints unix user:

```
~/bin/generate_abstracts <your_eprints_archive_name>
```

Also the eprints browse views will need to be regenerated to include the new seo08 and for08 browse views. Execute the following command, as the Eprints unix user:

```
~/bin/generate_views <your_eprints_archive_name>
```

5.3 Mapping Statistics

The eprint_seo mysql table contains the old SEO 1998 codes assigned to all Eprints. This table contained 3091 rows just prior to the running of the automatic seo1998 to seo2008 translation script. The eprint_seo08 mysql table contained 1794 entries after this script was run. 58% of all Eprints SEO codes were successfully mapped using this automatic one-to-one mapping process.

The eprint_subjects mysql table contains all the old RFCD 1998 codes assigned to all Eprints. This table contained 9018 rows just prior to the running of the automatic rfcd1998 to for2008 translation script. The eprint_seo08 mysql table contained 4124 entries after this script was run. 45% of all Eprints RFDC codes were successfully mapped using this automatic one-to-one mapping process. However because the old RFDC implementation incorrectly allowed codes that end in “00” to be selected, we have 2497 of the original 9018 that cannot be mapped at all. If we remove these from the equation, 4124 of a possible 6521 classifications were mapped – a successful mapping of 63%.

| Mapping | Total Number of Assigned Codes | Total Number of Assigned Codes Mapped | Percentage of Assigned Codes Mapped |
|----------------------|--------------------------------|---------------------------------------|-------------------------------------|
| SEO 1998 / FOR 2008 | 3091 | 1794 | 58% |
| RFCD 1998 / FOR 2008 | 6521 | 4124 | 63% |

6. References, Acknowledgements and Additional Resources

Australian Bureau of Statistics ANZSRC 2008 Classification:

<http://www.abs.gov.au/AUSSTATS/abs@.nsf/Latestproducts/1297.0Contents12008?opendocument&tabna>

Institutional Repositories Community (Australia/New Zealand) Google Group

<http://groups.google.com/group/institutionalrepositoriescommunity-anz?hl=en>

(Note: you must apply for group membership)

Lane, Bernard (2008) *Classification regime revealed*, The Australian, April 01, 2008.

<http://www.theaustralian.news.com.au/story/0,,23459308-12332,00.html>

LibraryFIT Blog (QUT):

<http://libraryfit.wordpress.com/2008/04/15/all-about-the-anzsrc/>

Minister - Innovation, Industry, Science and Research Web Site:

<http://www.minister.industry.gov.au/SenatortheHonKimCarr/Pages/NEWCLASSIFICATIONTOASSISTERADEVELOPMENT.aspx>

Research Administrators Wiki – Research Codes

http://www.radwiki.edu.au/index.php/Research_codes

Working Party (Eprints) Meeting Minutes and Agendas

- Members of the Eprints Working Party as of 8th of July:
 - Davies, Lynn (UTas Library)
 - Delphin, Tracey (UTas Library)
 - McCoy, Meredith (UTas Research Office / UTas Library)
 - McGee, Christian (School of Computing and Information Systems)
 - Sale, Arthur (School of Computing and Information Systems)
 - Rowlands, Derek (UTas Library)

Appendix A: MySQL commands required to implement ANZSRC 2008

A1: File Listing – for08.sql

```
DROP TABLE IF EXISTS `eprint_for08`;
CREATE TABLE `eprint_for08` (
  `eprintid` int(11) default NULL,
  `pos` int(11) default NULL,
  `for08` varchar(255) default NULL,
  KEY `eprintid` (`eprintid`),
  KEY `pos` (`pos`),
  KEY `for08` (`for08`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
alter table eprint__ordervalues_en add for08 text default
NULL after subjects;
```

A2: File Listing – seo08.sql

```
DROP TABLE IF EXISTS `eprint_seo08`;
CREATE TABLE `eprint_seo08` (
  `eprintid` int(11) default NULL,
  `pos` int(11) default NULL,
  `seo08` varchar(255) default NULL,
  KEY `eprintid` (`eprintid`),
  KEY `pos` (`pos`),
  KEY `seo08` (`seo08`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
alter table eprint__ordervalues_en add seo08 text default
NULL after for08;
```

Appendix B: Perl scripts sed to convert 2008 SEO/FOR txt files to Eprints XML

B1: Program Listing - parse_for.pl

```
#!/usr/bin/perl

$FOR_TXT = "./FOR2008.txt";
$FOR_XML = "./FOR2008.xml";

open for_txt_fh, "<$FOR_TXT";
open for_xml_fh, ">$FOR_XML";

my @subject_root;
$subject_root[0] = "ROOT";

my $prevlevel = 0;
my $prevsubject = "root";

print for_xml_fh "<?xml version=\"1.0\" encoding=\"utf-8\" ?>\n";
print for_xml_fh "<subjects>\n";

while ($line = <for_txt_fh>)
{
    chomp $line;
    ($level, $subjectid, $description) = split /\t/, $line;
    if ($level > $prevlevel)
    {
        $subject_root[$level] = $prevsubject;
    }
    print for_xml_fh "    <subject xmlns=\"http://eprints.org/ep2/data/2.0\">\n";
    print for_xml_fh "        <subjectid>$subjectid</subjectid>\n";
    print for_xml_fh "        <rev_number>1</rev_number>\n";
    print for_xml_fh "        <name>\n";
    print for_xml_fh "            <item>\n";
    if ($level == 0)
    {
        print for_xml_fh "                <name>$description</name>\n";
    }
    else
    {
        print for_xml_fh "                <name>$subjectid $description</name>\n";
    }
    print for_xml_fh "            <lang>en</lang>\n";
    print for_xml_fh "        </item>\n";
    print for_xml_fh "    </name>\n";
    print for_xml_fh "    <parents>\n";
    print for_xml_fh "        <item>" . $subject_root[$level] . "</item>\n";
    print for_xml_fh "    </parents>\n";
    if ($level == 3)
    {
        print for_xml_fh "        <depositable>TRUE</depositable>\n";
    }
    else
    {
        print for_xml_fh "        <depositable>FALSE</depositable>\n";
    }
    print for_xml_fh "    </subject>\n";
    $prevlevel = $level;
    $prevsubject = $subjectid;
}

print for_xml_fh "</subjects>\n";

close for_txt_fh;
close for_xml_fh;
```

B2: Program Listing - parse_seo.pl

```
#!/usr/bin/perl

$SEO_TXT = "./SEO2008.txt";
$SEO_XML = "./SEO2008.xml";

open seo_txt_fh, "<$SEO_TXT";
open seo_xml_fh, ">$SEO_XML";

my @subject_root;
$subject_root[0] = "ROOT";

my $prevlevel = 0;
my $prevsubject = "root";

print seo_xml_fh "<?xml version=\"1.0\" encoding=\"utf-8\" ?>\n";
print seo_xml_fh "<subjects>\n";

while ($line = <seo_txt_fh>)
{
    chomp $line;
    ($level, $subjectid, $description) = split /\t/, $line;
    if ($level > $prevlevel)
    {
        $subject_root[$level] = $prevsubject;
    }
    print seo_xml_fh "    <subject xmlns=\"http://eprints.org/ep2/data/2.0\">\n";
    print seo_xml_fh "        <subjectid>$subjectid</subjectid>\n";
    print seo_xml_fh "        <rev_number>1</rev_number>\n";
    print seo_xml_fh "        <name>\n";
    print seo_xml_fh "            <item>\n";
    if ($level == 0)
    {
        print seo_xml_fh "                <name>$description</name>\n";
    }
    else
    {
        print seo_xml_fh "                <name>$subjectid $description</name>\n";
    }
    print seo_xml_fh "            <lang>en</lang>\n";
    print seo_xml_fh "        </item>\n";
    print seo_xml_fh "    </name>\n";
    print seo_xml_fh "    <parents>\n";
    print seo_xml_fh "        <item>" . $subject_root[$level] . "</item>\n";
    print seo_xml_fh "    </parents>\n";
    if ($level == 4)
    {
        print seo_xml_fh "        <depositable>TRUE</depositable>\n";
    }
    else
    {
        print seo_xml_fh "        <depositable>FALSE</depositable>\n";
    }
    print seo_xml_fh "    </subject>\n";
    $prevlevel = $level;
    $prevsubject = $subjectid;
}

print seo_xml_fh "</subjects>\n";

close seo_txt_fh;
close seo_xml_fh;
```

Appendix C: Perl scripts to implement the ASRC1998 to ANZSRC2008 Mapping

C1: Program Listing - Settings.pm

```
package Settings;

# Toggle updating of database ON (1) or OFF (0)
our $UPDATEDB = 0;

# MySQL database connection details
our $DATABASE = "<your_eprints_db_dbname>";
our $DBUSER = "<your_eprints_db_username>";
our $DBPASSWORD = "<your_eprints_db_password>";
our $DBHOST = "<your_eprints_db_host>";

# Path Names of Translation Files
our $FOR_FILE = "./for2008.csv";
our $SEOS_FILE = "./seo2008.csv";

1;
```

C2: Program Listing - Functions.pm

```
package Functions;

use Settings;

sub connect_database
{
    my $database = $Settings::DATABASE;
    my $dbhost = $Settings::DBHOST;
    my $dbuser = $Settings::DBUSER;
    my $dbpassword = $Settings::DBPASSWORD;
    my $dsn = "DBI:mysql:database=$database;host=$dbhost";
    my $dbh = DBI->connect($dsn, $dbuser, $dbpassword) or die "ERR\n";
    return $dbh;
}

sub read_translations
{
    {
        $filename = $_[0];
        %trans = ();
        open (TRANS, $filename);
        while (<TRANS>)
        {
            my $line = $_;
            $line =~ s/[\n\r]//g;
            my ($class98, $class08) = split(/,/, $line);
            if (defined $trans{$class98})
            {
                $trans{$class98} = $trans{$class98} . ":" . $class08;
            }
            else
            {
                $trans{$class98} = $class08;
            }
        }
        return %trans;
    }
}

sub one_to_one
{
    my $class = ${$_[0]};
    my %one2one = %{$_[1]};
    my $dbh = ${$_[2]};
    my $dbtable = "eprint_" . $class;
    print "\n";
    print "===== \n";
    print "| One to One Translations (handled automatically) | \n";
    print "| eprintID: $class | \n";
    print "===== \n \n";
}
```

```

foreach $id (sort { $a <=> $b } keys %one2one)
{
    my @newsobjects = ();
    @newsobjects = split(/:/, $one2one{$id});
    my $numnewsobjects = scalar(@newsobjects);
    print $id . ": ";
    for (my $i = 0; $i < $numnewsobjects; $i++)
    {
        print $newsobjects[$i] . " ";
        if ($Settings::UPDATEDB)
        {
            my $statement = "INSERT INTO $dbtable VALUES
                ('$id', '$i', '$newsobjects[$i]')";
            my $query = $dbh->prepare($statement);
            $query->execute
            or die "unable to execute query: $dbh->errorstr\n";
            $query->finish;
        }
    }
    print "\n";
}

sub one_to_many
{
    my $class = ${$_[0]};
    my %one2many = %{$_[1]};
    print "\n";
    print "===== \n";
    print "| One to Many Translations (handled manually) | \n";
    print "| eprintID / old subject: new $class | \n";
    print "===== \n\n";
    foreach $id (sort { $a <=> $b } keys %one2many)
    {
        $subjects = $one2many{$id};
        foreach $subject (sort { $a <=> $b } keys %$subjects)
        {
            my @newsobjects = ();
            @newsobjects = split(/:/, $$subjects{$subject});
            my $numnewsobjects = scalar(@newsobjects);
            print $id . " / " . $subject . ": ";
            for (my $i = 0; $i < $numnewsobjects; $i++)
            {
                my $for = substr $newsobjects[$i], 0, 6;
                print $for . " ";
            }
            print "\n";
        }
    }
}

sub non_depositable
{
    my $class = ${$_[0]};
    my %notdepositable = %{$_[1]};
    print "\n";
    print "===== \n";
    print "| Non-depositable subjects | \n";
    print "| eprintID: subjects | \n";
    print "===== \n\n";
    foreach $id (sort { $a <=> $b } keys %notdepositable)
    {
        my @subjects = ();
        @subjects = split(/:/, $notdepositable{$id});
        my $numsubjects = scalar(@subjects);
        print "$id: ";
        for (my $i = 0; $i < $numsubjects; $i++)
        {
            my $subject = substr $subjects[$i], 0, 6;
            print $subject . " ";
        }
        print "\n";
    }
}

1;

```

C3: Program Listing - for1998-2-2008.pl

```
#!/usr/bin/perl

#####
#       for1998-2-2008.pl
#####

use DBI;
use Functions;
use Settings;
my %one2one = ();
my %one2many = ();
my %notdepositable = ();
my %trans_table = ();
%trans_table = Functions::read_translations($Settings::FOR_FILE);
#connect to the database
$dbh = Functions::connect_database();
my $query = $dbh->prepare("SELECT * FROM eprint_subjects ORDER BY subjects, pos");
$query->execute or die "unable to execute query: $dbh->errorstr\n";
$numRows = $query->rows;
while (my $ref = $query->fetchrow_arrayref) {
    $eprintid = $$ref[0];
    $pos = $$ref[1];
    $subject = $$ref[2];
    $new = substr $trans_table{$subject}, 0, 6;
    $new_last2digits = substr $new, 4, 2;
    $subject_last2digits = substr $subject, 4, 2;
    if ((!($trans_table{$subject} =~ ":") && ($new ne ""))
        && ($new_last2digits ne "99") && ($subject_last2digits ne "99"))
    {
        if (defined $one2one{$eprintid})
        {
            $one2one{$eprintid} = $one2one{$eprintid} . ":" . $new;
        }
        else
        {
            $one2one{$eprintid} = $new;
        }
    }
    else
    {
        # Set up lists of eprintIDs and subjects/new subjects to manually check
        if (($trans_table{$subject} =~ ":") || ($new_last2digits eq "99")
            || ($subject_last2digits eq "99"))
        {
            if (defined $one2many{$eprintid}{$subject})
            {
                $one2many{$eprintid}{$subject} =
                    $one2many{$eprintid}{$subject} . ":" .
                    $trans_table{$subject};
            }
            else
            {
                $one2many{$eprintid}{$subject} = $trans_table{$subject};
            }
        }
        if ($new eq "")
        # old subject should be non-depositable
        {
            if (defined $notdepositable{$eprintid})
            {
                $notdepositable{$eprintid} = $notdepositable{$eprintid} .
                    ":" . $subject;
            }
            else
            {
                $notdepositable{$eprintid} = $subject;
            }
        }
    }
}
$query->finish;
$class = "for08";
&Functions::one_to_one($class, \%one2one, $dbh);
&Functions::one_to_many($class, \%one2many);
&Functions::non_depositable($class, \%notdepositable);
$dbh->disconnect;
exit;
```

C4: Program Listing - seo1998-2-2008.pl

```
#!/usr/bin/perl

#####
#      seo1998-2-2008.pl
#####

use DBI;
use Functions;
use Settings;
my %one2one = ();
my %one2many = ();
my %notdepositable = ();
my %trans_table = ();
%trans_table = Functions::read_translations($Settings::SEOS_FILE);
#connect to the database
$dbh = Functions::connect_database();
my $query = $dbh->prepare("SELECT * FROM eprint_seos ORDER BY seos, pos");
$query->execute or die "unable to execute query: $dbh->errorstr\n";
$numRows = $query->rows;
while (my $ref = $query->fetchrow_arrayref) {
    $eprintid = $$ref[0];
    $pos = $$ref[1];
    $subject = $$ref[2];
    $new = substr $trans_table{$subject}, 0, 6;
    $new_last2digits = substr $new, 4, 2;
    $subject_last2digits = substr $subject, 4, 2;
    if ((!($trans_table{$subject} =~ ":") && ($new ne ""))
        && ($new_last2digits ne "99") && ($subject_last2digits ne "99"))
    {
        if (defined $one2one{$eprintid})
        {
            $one2one{$eprintid} = $one2one{$eprintid} . ":" . $new;
        }
        else
        {
            $one2one{$eprintid} = $new;
        }
    }
    else
    {
        # Set up lists of eprintIDs and subjects/new subjects to manually check
        if (($trans_table{$subject} =~ ":") || ($new_last2digits eq "99")
            || ($subject_last2digits eq "99"))
        {
            if (defined $one2many{$eprintid}{$subject})
            {
                $one2many{$eprintid}{$subject} =
                    $one2many{$eprintid}{$subject} . ":" .
                    $trans_table{$subject};
            }
            else
            {
                $one2many{$eprintid}{$subject} = $trans_table{$subject};
            }
        }
        if ($new eq "")
        # old subject should be non-depositable
        {
            if (defined $notdepositable{$eprintid})
            {
                $notdepositable{$eprintid} = $notdepositable{$eprintid} .
                    ":" . $subject;
            }
            else
            {
                $notdepositable{$eprintid} = $subject;
            }
        }
    }
}
$query->finish;
$class = "seo08";
&Functions::one_to_one($class, \%one2one, $dbh);
&Functions::one_to_many($class, \%one2many);
&Functions::non_depositable($class, \%notdepositable);
$dbh->disconnect;
exit;
```